Introduction:

Robust and High Performance Tools for Scientific Computing

Tony Drummond
Lawrence Berkeley National Laboratory
ACTS Collection Workshop
August 5-8, 2003

Motivation

Grand Challenges are ..fundamental problems in science and engineering, with potentially broad social, political, and scientific impact, that could be advanced by applying high performance computer resources

Office of Science and Technology

- Some grand challenges: electronic structure of materials, turbulence, genome sequencing and structural biology, global climate modeling, speech and language studies, pharmaceutical design, pollution, etc. .

Motivation

With the development of new kinds of equipment of greater capacity, and particularly of greater speed, it is almost certain that new methods will have to be developed in order to make the fullest use of this equipment. It is necessary not only to design machines for the mathematics, but also to develop a new mathematics for the machines - 1952, Hartree

- **Metropolis** grew out of physical chemistry in 1950's through attempts to calculate statistical properties of chemical reactions. Some areas of application: astrophysics, many areas engineering, and chemistry
- **Fast Fourier Transform** (**FFT**): implementation of Fourier Analysis. Some areas of application: image and signal processing, seismology, physics, radiology, acoustics and engineering
- **Multigrids**: method for solving a wide variety of PDE. Some areas of application: physics, biophysics and engineering

Motivation

Computational science: can be characterized by the needs to gain understanding through the analysis of mathematical models using high performing performing computers

Community

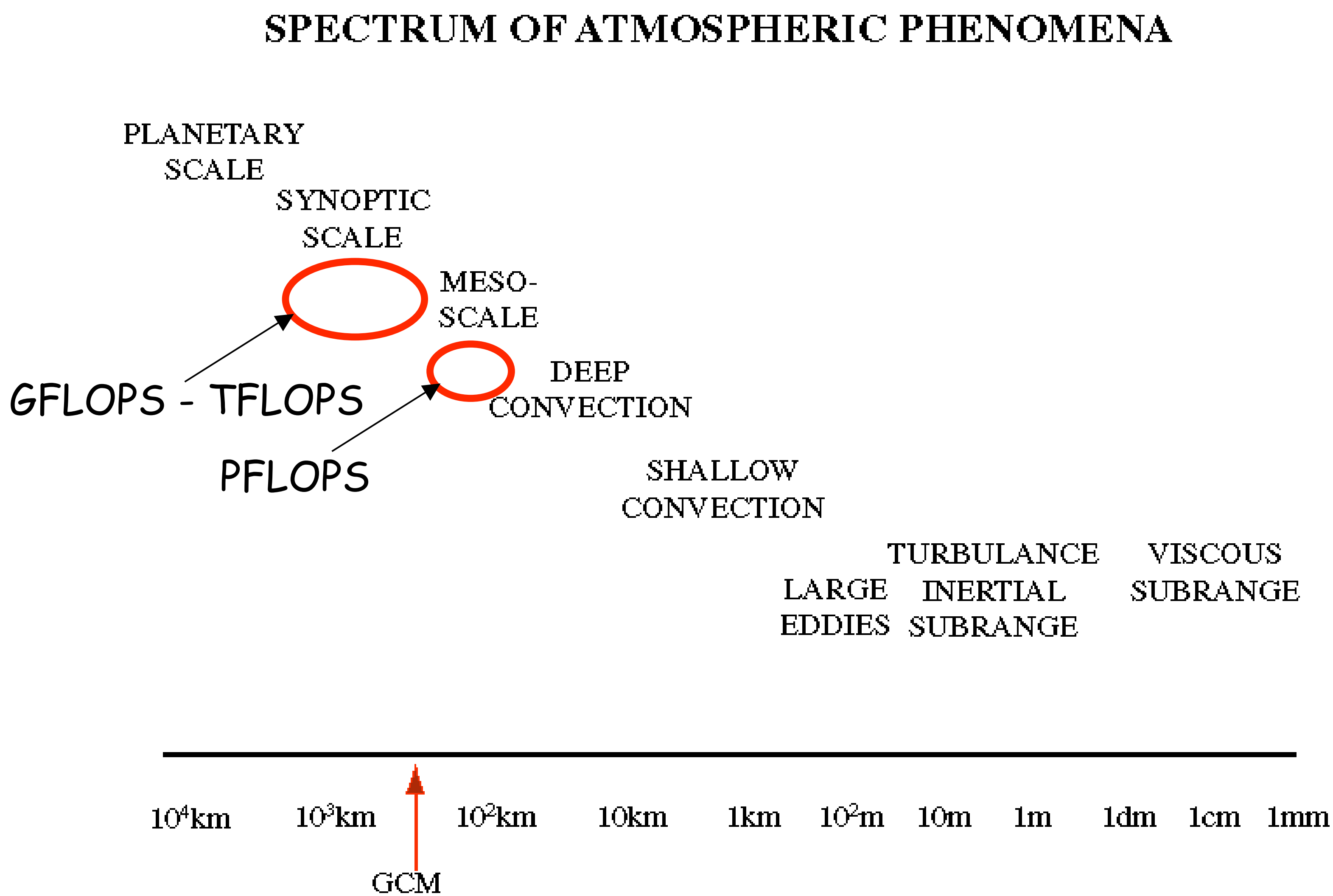
- Scientists
- Engineers
- Mathematicians
- Economists, artists

Multidisciplinary!

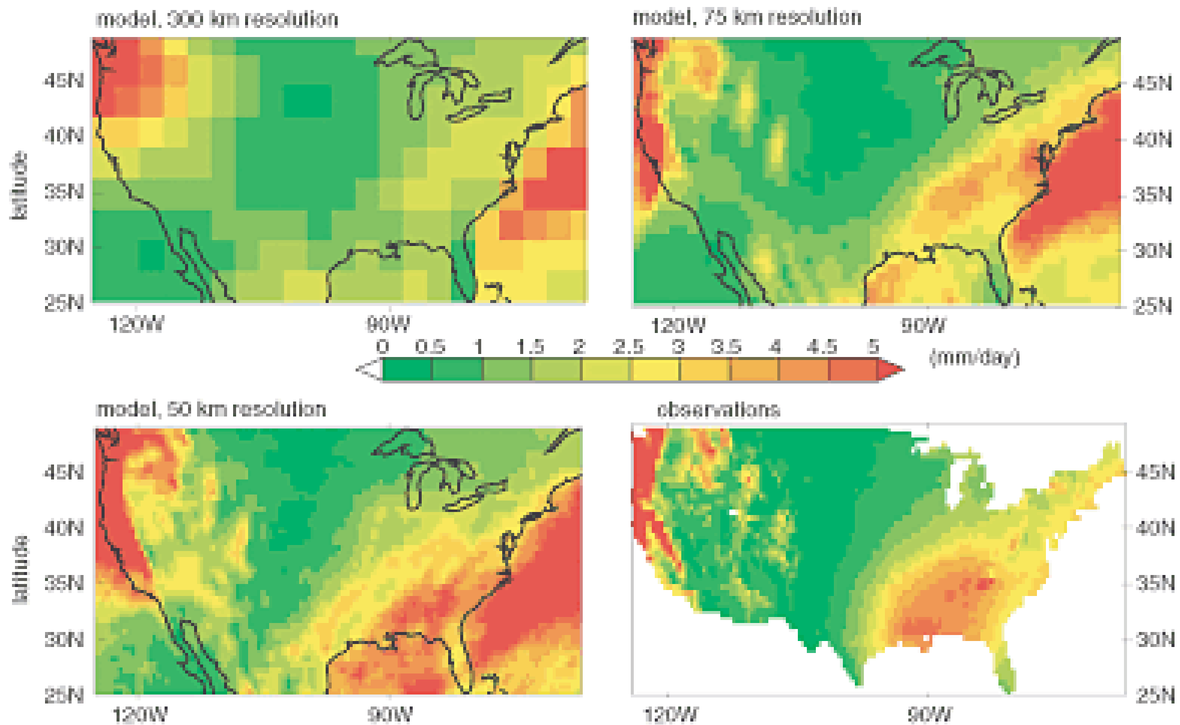
Computer Science	Mathematics:
Provides services ranging from Networking and visualization tools to algorithms	Credibility of algorithms (error analysis, exact solutions, expansions, uniqueness proofs and theorems)

Some lessons learned from Earth System Modeling

Motivation - Example I



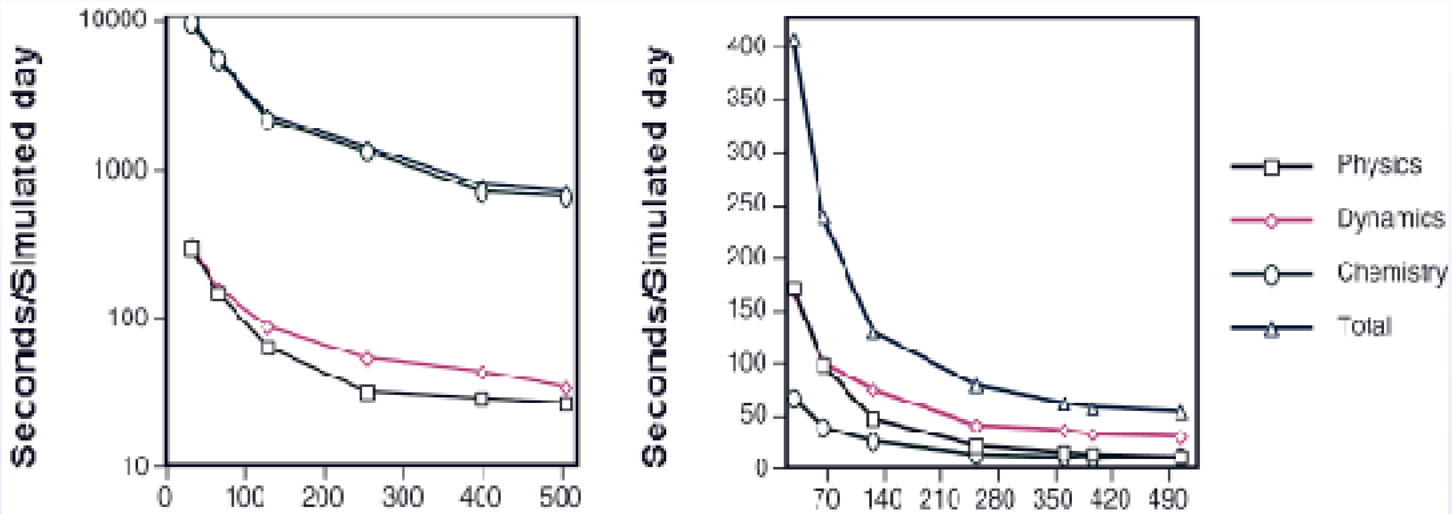
Motivation - Example I



Duffy et. al.,
Lawrence Livermore National Laboratory

- CCM3 - spectral truncations of T170 and T239
- 50 Km spatial resolution is 32 times more grid cells and takes roughly **200 times longer** to run

Motivation - Example II



AGCM/ACM
2.5 long x 2 lat, 30 layers
25-chemical species

AGCM/ACM
2.5 long x 2 lat, 30 layers
2-chemical species

- **Non-linear demand for resources (CPU - Memory)**
- **Multi-disciplinary application is more demanding**

Using today's hardware to tackle today's Grand Challenges

Q. Why is it **still** difficult to obtain High Performance?

Some common and *interesting* answers

- Technology
- Memory latency
- Algorithms
- Programming Practices

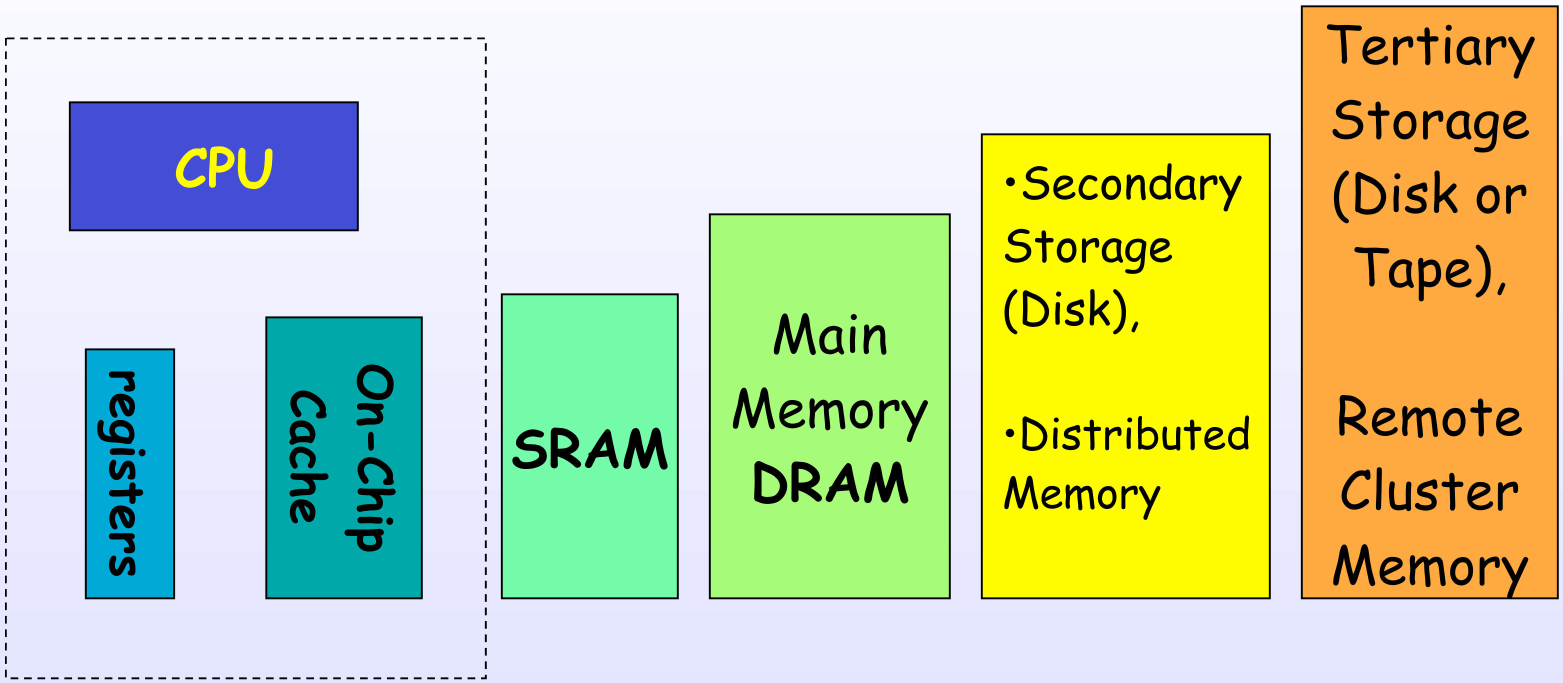
⋮

Some options for New Architectures

OPTION	SOFTWARE IMPACT	COST	TIMELINE
Modification of commodity processors	Minimal	2 or 3 times commodity?	Can be achieved in a few years
U.S.-made vector architecture	Moderate	2 or 3 times commodity at present	Available now
Processor-in-memory (Blue Gene/L)	Extensive	Unknown, 2 to 5 times commodity?	Only prototypes available now
Japanese- made vector architecture	Moderate	2.5 to 3 times commodity at present	Available now
Research Architectures (Streams, VIRAM..)	Extensive or unknown	Unknown	Academic research prototypes only available now

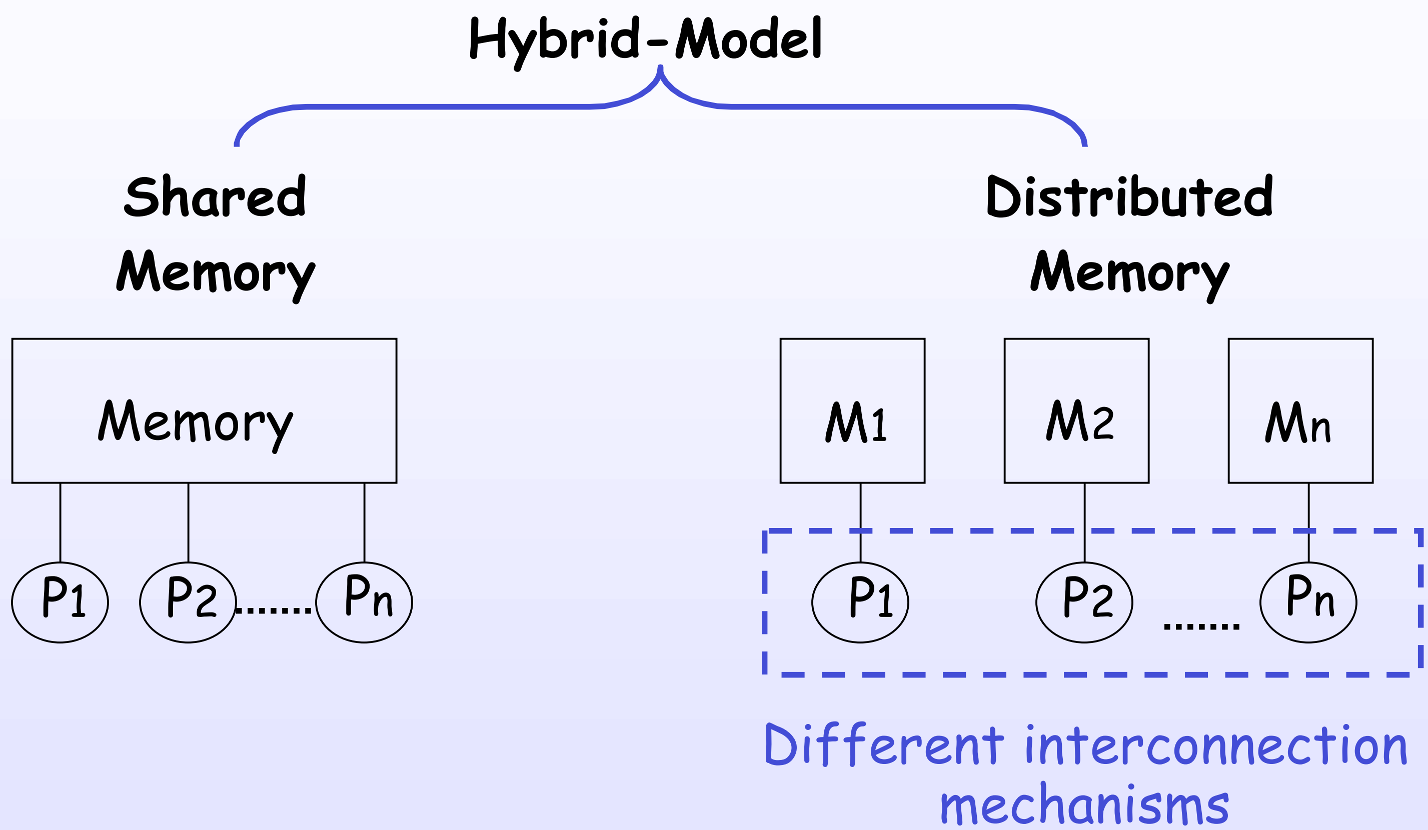
Memory Hierarchy

- *Where is the data? Why is data locality important?*



Speed	1's ns	10's ns		100's ns	.1's -10's ms	10's s
Size	100's bytes	Kbytes		Mbytes	Gbytes	Tbytes

Memory Latency



The GRID

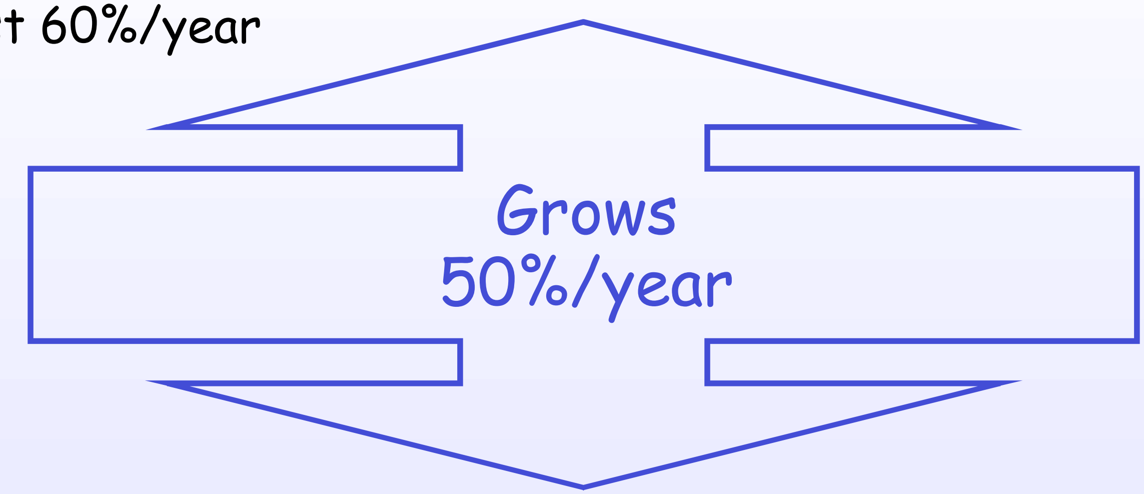
- A large pool of resources
  - Computers
  - Networks
  - Software
  - Databases
  - Instruments
  - people

Requirements from GRID implementation:

- Ubiquitous: ability to interface to the grid at any point and leverage whatever is available
- Resource Aware: manage heterogeneity of resources
- Adaptive: tailored to obtain maximum performance from resources

CPU vs. DRAM Performance

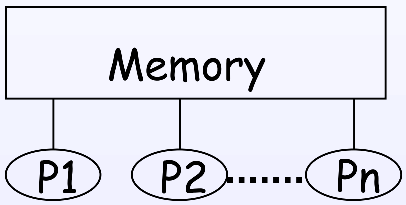
- Since 1980's, µProcs performance has increased at a rate of almost 60%/year



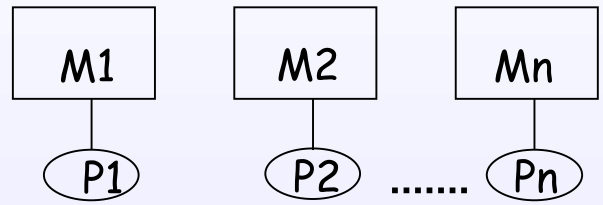
- Since 1980's, DRAM (latency) has improved at a rate of almost 9%/year

Parallel Programming Paradigms

Shared Memory



Distributed Memory



- Data parallelism

Shared memory:
- easier to implement
- shared memory space
- mutual exclusion, contention

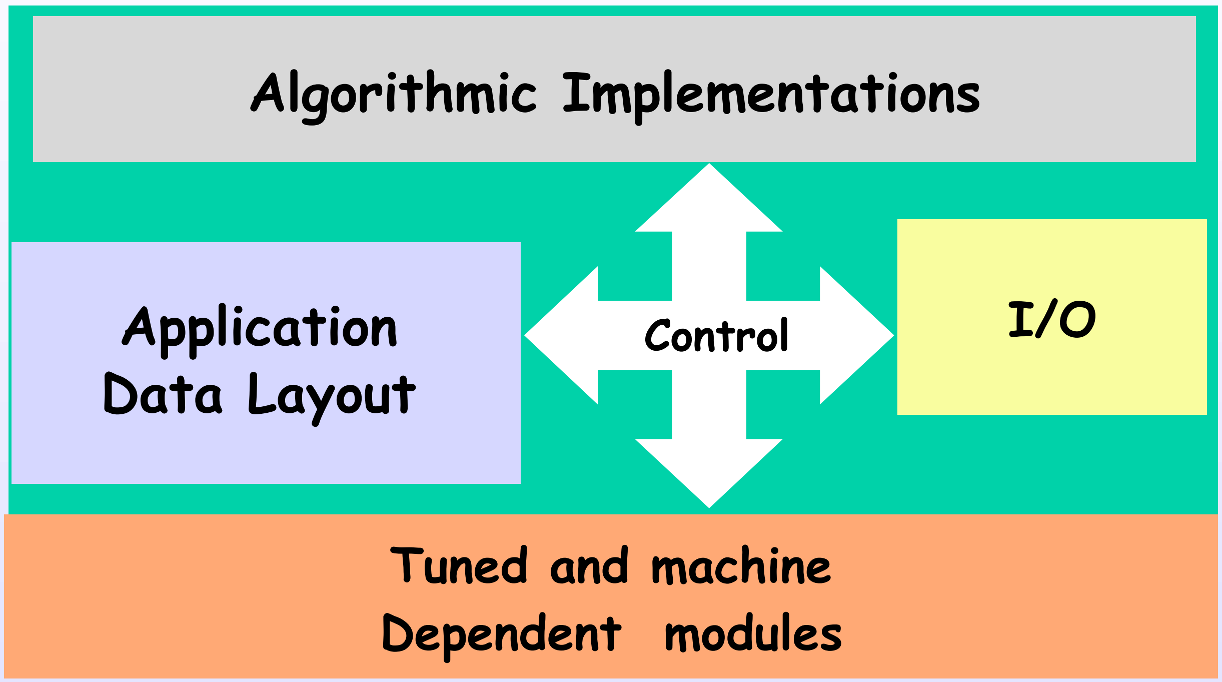
Distributed memory:
- virtual shared memory
- data is implicitly available to all

- Message Passing

Shared memory:
- shared area is use for sending and receiving data

Distributed memory:
- Implicit mutual exclusion
- Only explicit synch
- Depends on Memory Hierarchy and Network

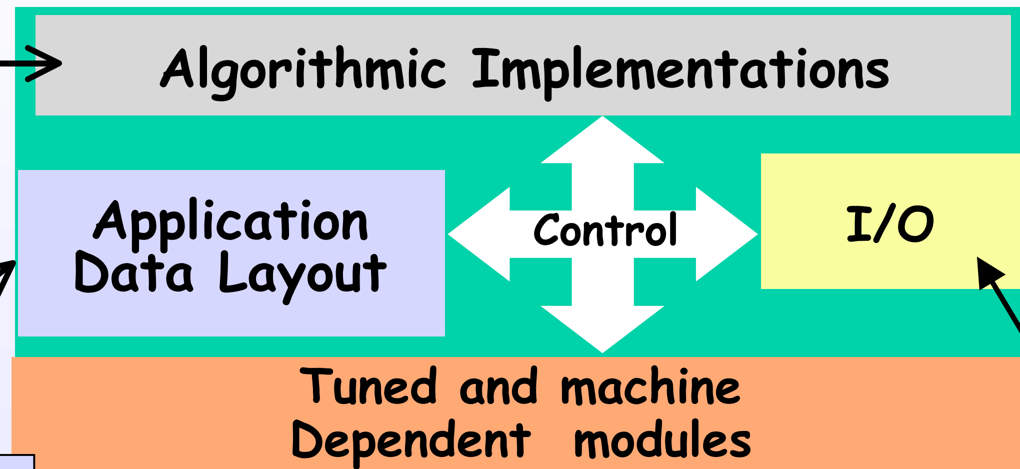
Large Scientific Codes:
A Common Programming Practice



Shortcomings

New Architecture:
- May or may not need re-rewriting

New Developments:
- Difficult to compare



New Architecture:
- Extensive re-rewriting

New or extended Physics:
- Extensive re-rewriting or increase overhead

New Architecture:
- Minimal to Extensive rewriting

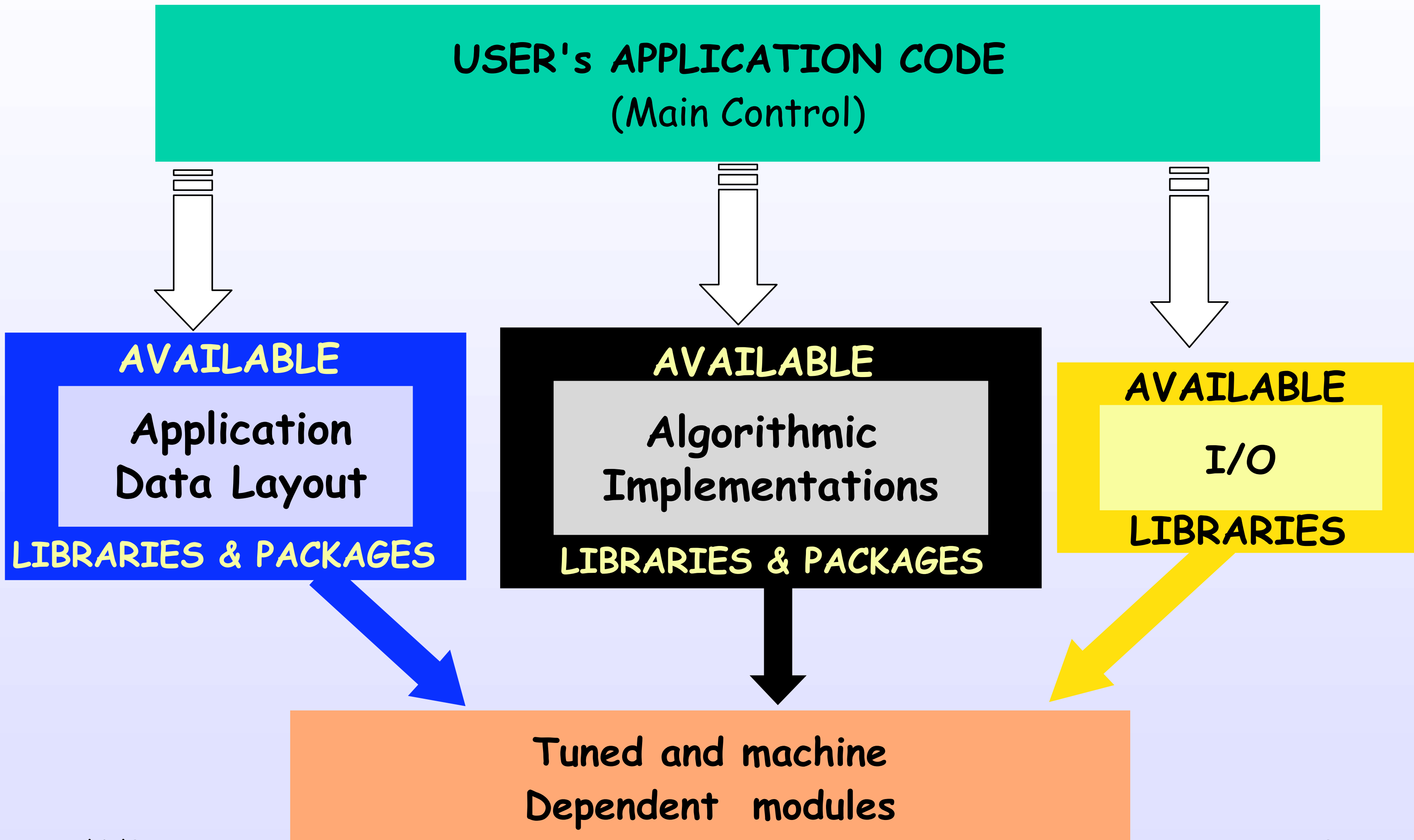
New Architecture or S/W
- Extensive tuning
- May require new programming paradigms
- **Difficult to maintained!**

Shortcomings...?

"We need to move away from a coding style suited for serial machines, where every macrostep of an algorithm needs to be thought about and explicitly coded, to a higher-level style, where the compiler and library tools take care of the details. And the remarkable thing is, if we adopt this higher-level approach right now, even on today's machines, we will see immediate benefits in our productivity."

W. H. Press and S. A. Teukolsky, 1997
Numerical Recipes: Does This Paradigm Have a future?

Alternative Programming Approach



Software Development
Levels of abstraction

- Scientific or engineering context
- Domain expertise

- Simulation codes
- Data Analysis codes

- Templates
- Scientific Computing Tools
- General Purpose Libraries

- Data Structures
- Algorithms
- Code Optimization
- Programming Languages
- O/S - Compilers

Hardware - Middleware - Firmware

Partial Matrix of Methods and Disciplines

	Climate Change	Material Science	High Enregy Physics	Astrophysics Cosmology	Biology	Chemistry	Fusion
Monte Carlo (Quantum and Classical)	PCM CCSM POP	Quantum MC Classical KMC	FASTER SYNPOL	FASTER SYNPOL		NWChem	
Fast Fourier Transform		VASP Paratec Petot Escan	IMPACT LANGEVIN3D MAD9P ccSHT		SPIDER NAMD	NAMD	WARP GTC
Fast Multipole & Variants		Classical MD	IMPACT LANGEVIN3D QuickPIC		Classical MD	NWChem Classical MD	
Sparse Linear systems	PCM CCSM POP	O(N) Methods	OMEGA3P		SPIDER	pVarDen	NIMROD
Eigenvalue Solvers		DFT FLAPW PW codes	OMEGA3P		DFT SPIDER	NWChem Gaussian QChem	
Dense Linear Solvers		LSMS FLAPW		MADCAP		NWChem Gaussian	GTC
Adaptive Mesh Refinement	BoxLib Paramesh		BoxLib Paramesh	FLASH Paramesh		pVarDen BoxLib	WARP BOX Chombo

Partial Matrix of Methods and Disciplines

	Climate	Material	High Energy	Astrophysics	Biology	Chemistry	Fusion
Monte Carlo (Quantum, Classical)	POP					NWChem	
Fast Fourier Transform		VASP Paratec	IMPACT		SPIDER		WARP GTC
Fast Multipole & Variants		Classical MD	QuickPIC		MD	Classical MD	
Sparse Linear systems	PCM CCSM POP	O(N) Methods	OMEGA3P		SPIDER	pVarDen	NIMROD
Eigenvalue Solvers		DFT FLAPW PW codes				QChem	
Dense Linear Solvers		LSMS FLAPW		MADCAP		NWChem Gaussian	GTC
Adaptive Mesh Refinement	BoxLib Paramesh		BoxLib Paramesh	FLASH Paramesh		pVarDen BoxLib	WARP BOX Chombo

NWChem: Uses GlobalArrays for implementing distributing computing environments

MADCAP: Uses ScaLAPACK for the solution of large and dense linear systems of equations

FLAPW: uses ScaLAPACK for the solutions Dense Eigenvalue Problems

What is the DOE ACTS Collection?

http://acts.nersc.gov

- Advanced CompuTational Software
- Tools for developing parallel applications
  - Developed (primarily) at DOE Labs
  - Separate projects originally
  - ~ 20 tools
- ACTS is an "umbrella" project
  - Leverage numerous independently funded projects
  - Collect tools in a toolkit

ACTS: *Project Goals*

- Extended support for *experimental software*
- Make ACTS tools available on DOE computers
- Provide technical support (*acts-support@nersc.gov*)
- Maintain ACTS information center (*http://acts.nersc.gov*)
- Coordinate efforts with other supercomputing centers
- Enable large scale scientific applications
- Educate and train

Related Activities

- **Software Repositories**:
  - Netlib: *http://www.netlib.org*
  - HPC-Netlib: *http://www.nhse.org/hpc-netlib*
  - National HPCC Software Exchange NHSE: *http://www.nhse.org*
  - Guide to Available Mathematical Software: *http://gams.nist.gov*
  - MGNet: *http://www.mgnet.org*
  - NEOS: *http://www-fp.mcs.anl.gov/otc/Guide*
  - OO Numerics: *http://oonumerics.org/oon*
- **Portable timing routines, tools for debugging, compiler technologies:**
  - Ptools: *http://www.ptools.org*
  - Center for Programming Models for Scalable Parallel Computing: *http://www.pmodels.org*
- **Education:**
  - Computational Science Educational Project: *http://csep1.phy.ornl.gov*
  - UCB's Applications of Parallel Computers: *http://www.cs.berkeley.edu/~demmel/cs267_Spr99*
  - MIT's Applied Parallel Computing: *http://www.mit.edu/~cly/18.337*
  - Dictionary of algorithms, data structures and related definitions: *http://www.nist.gov/dads*

Why is ACTS unique?

- Extended support for tools
- Accumulates the expertise and user feedback on the use of the software tools and scientific applications that used them:
  - independent software evaluations
  - participation in the developer user groups e-mail list
  - presentation of a gallery of applications
  - leverage between tool developers and tool users
  - workshops and tutorials
  - tool classification
  - support

ACTS Information Center

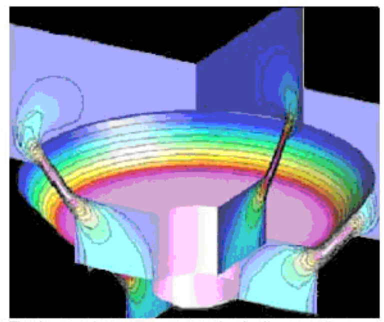
http://acts.nersc.gov

The DOE ACTS Collection

Tools | News | Project | Center | Search

The DOE ACTS (Advanced CompuTational Software) Collection is a set of DOE-developed software tools that make it easier for programmers to write high performance scientific applications for parallel computers. This site is the central information center for the ACTS Collection and is brought to you by NERSC and the Mathematical, Information, and Computational Sciences (MICS) Division of DOE. Correspondence regarding the collection (including requests for support) should be directed to acts-support@nersc.gov.

click on the image below to see other applications that have benefited from ACTS Tools



The image shows pressure and velocity around a moving valve in a diesel engine. The flow here was found as part of a CFD effort to simulate the flow within the complex 3D geometry of a diesel engine. The computation was carried out using the Overture Framework and the PADRE library for parallel data distribution.

Tools | News | Project | Center | Search

- Tools: Tool descriptions, installation details, examples, etc
- News: Agenda, accomplishments, conferences, releases, etc
- Project: Goals and other relevant information
- Center: Points of contact
- Search: Search engine

How much effort is involved in using these tools?

Using the ACTS Collection

When a tool is not available at your site. . .

- Download the tools (Freeware!)
- Most of the tools support many of the available computational platforms (even Windows!)
- Follow installation instructions (some tools provide "configuration scripts")

Using the ACTS Collection

- Most of the tools provide interfaces (calling functions and subroutines) from Fortran and C (some even C++)

```
 CALL BLACS_GET( -1, 0, ICTXT )
 CALL BLACS_GRIDINIT( ICTXT, 'Row-major', NPROW, NPCOL )
:
 CALL BLACS_GRIDINFO( ICTXT, NPROW, NPCOL, MYROW, MYCOL )
:
:
 CALL PDGESV( N, NRHS, A, IA, JA, DESCA, IPIV, B, IB, JB, DESCB,
 $                INFO )
```

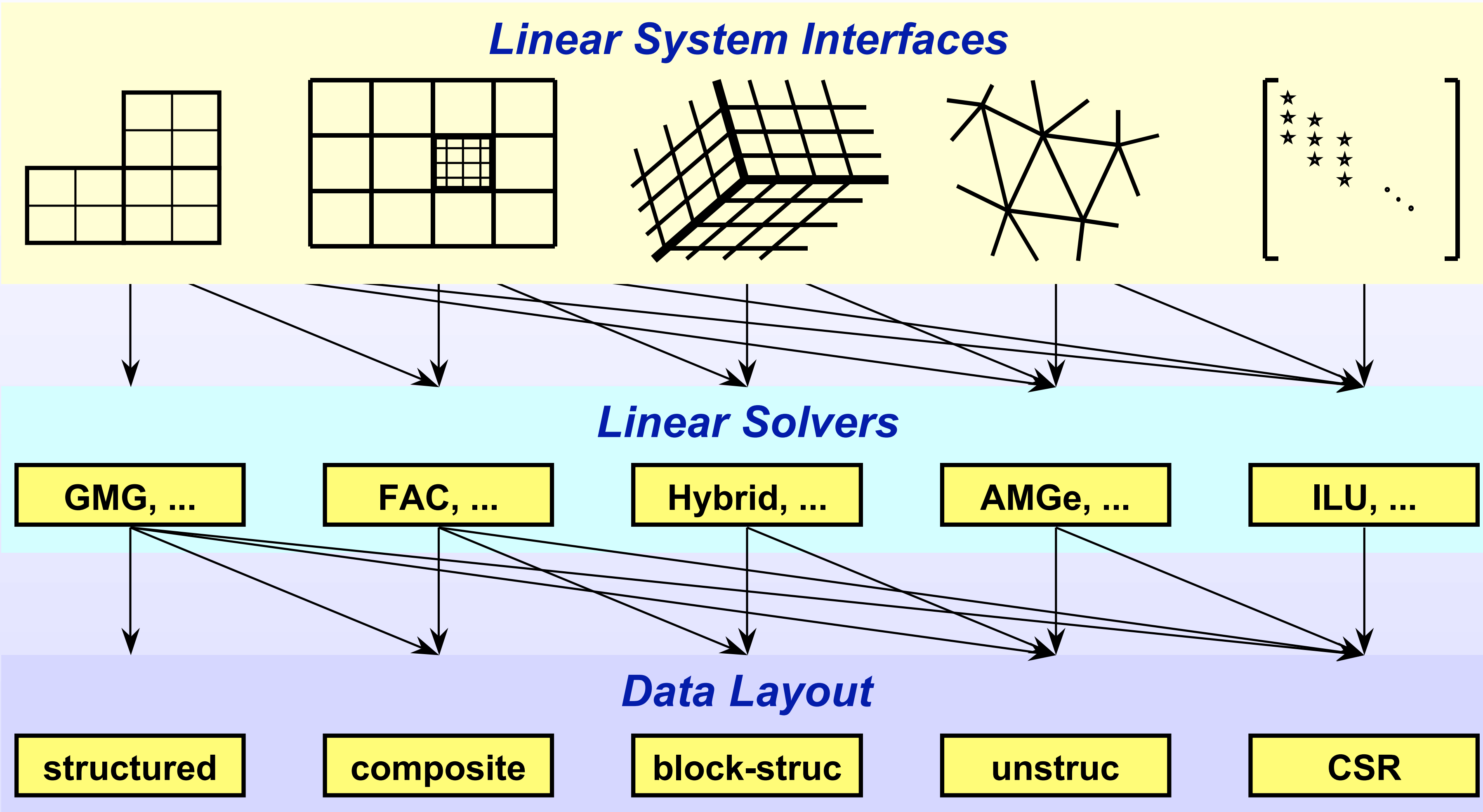
Using the ACTS Collection

- **-ksp_type [cg,gmres,bcgs,tfqmr,…]**
- **-pc_type [lu,ilu,jacobi,sor,asm,…]**

More advanced:

- **-ksp_max_it <max_iters>**
- **-ksp_gmres_restart <restart>**
- **-pc_asm_overlap <overlap>**
- **-pc_asm_type [basic,restrict,interpolate,none]**
- **Many more (see manual)**

Using the ACTS Collection



Using the ACTS Collection

- Best approach is to start with examples for beginners!
- Several efforts are targeting Tool Interoperability!

What needs to be computed?

ScaLAPACK

Aztec/Trilinos

SuperLU

$$Ax = b$$

$$Az = \lambda z$$

$$A = U \Sigma V^T$$

$$\min\left\{ \tfrac{1}{2} \| r(x) \|^2 : x_l \le x \le x_u \right\}$$

PDEs

ODEs

OPT++

TAO

PETSc

Hypre

SUNDIALS

What codes are being developed?

Global Arrays

PAWS

Overture

Parallel programs that use large distributed arrays

Coupling distributed applications

Support for Grids and meshes

Infrastructure for distributed computing

On-line visualization and computational stearing

Language Interoperability

Performance analysis and monitoring

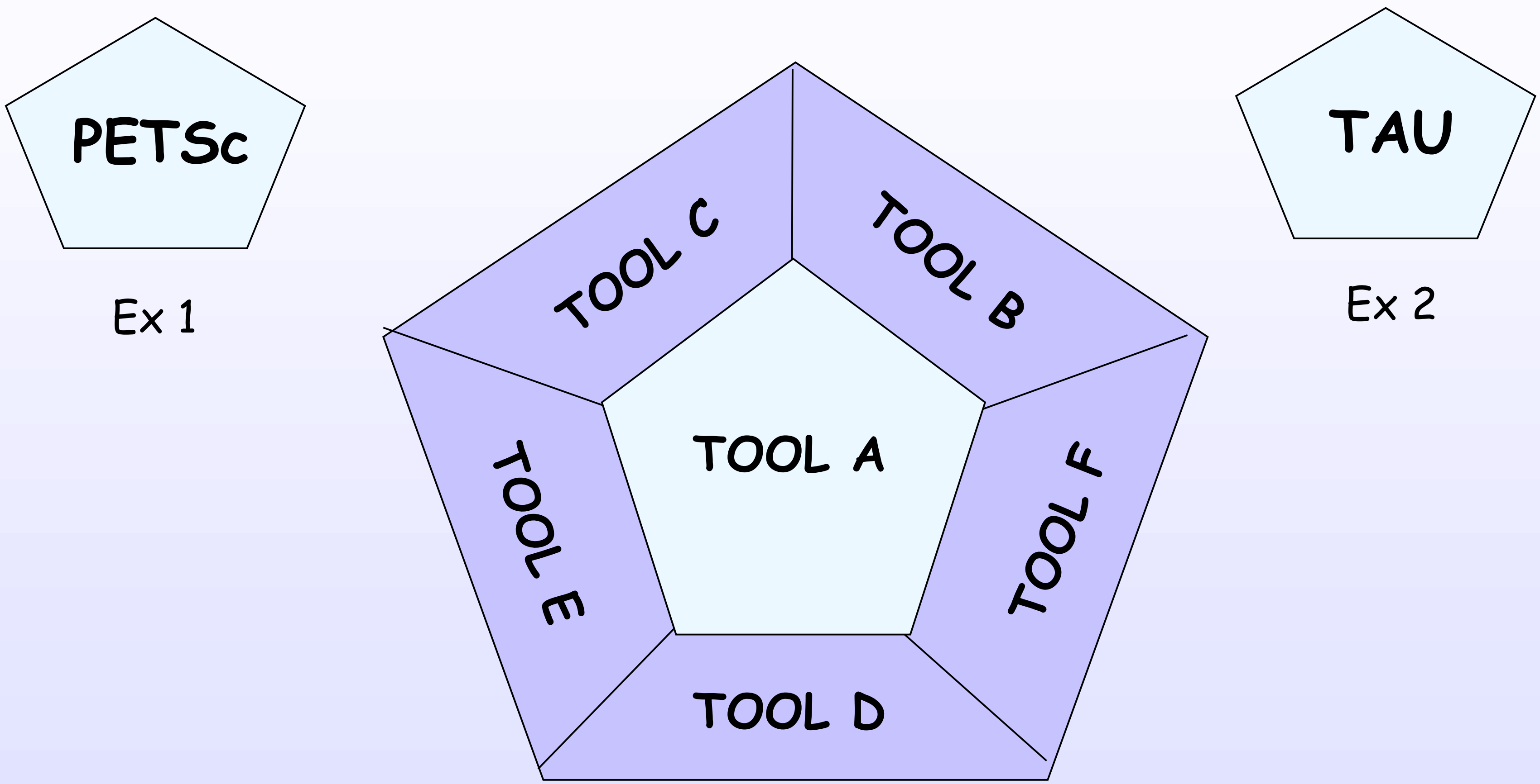
Chasm

CUMULVS

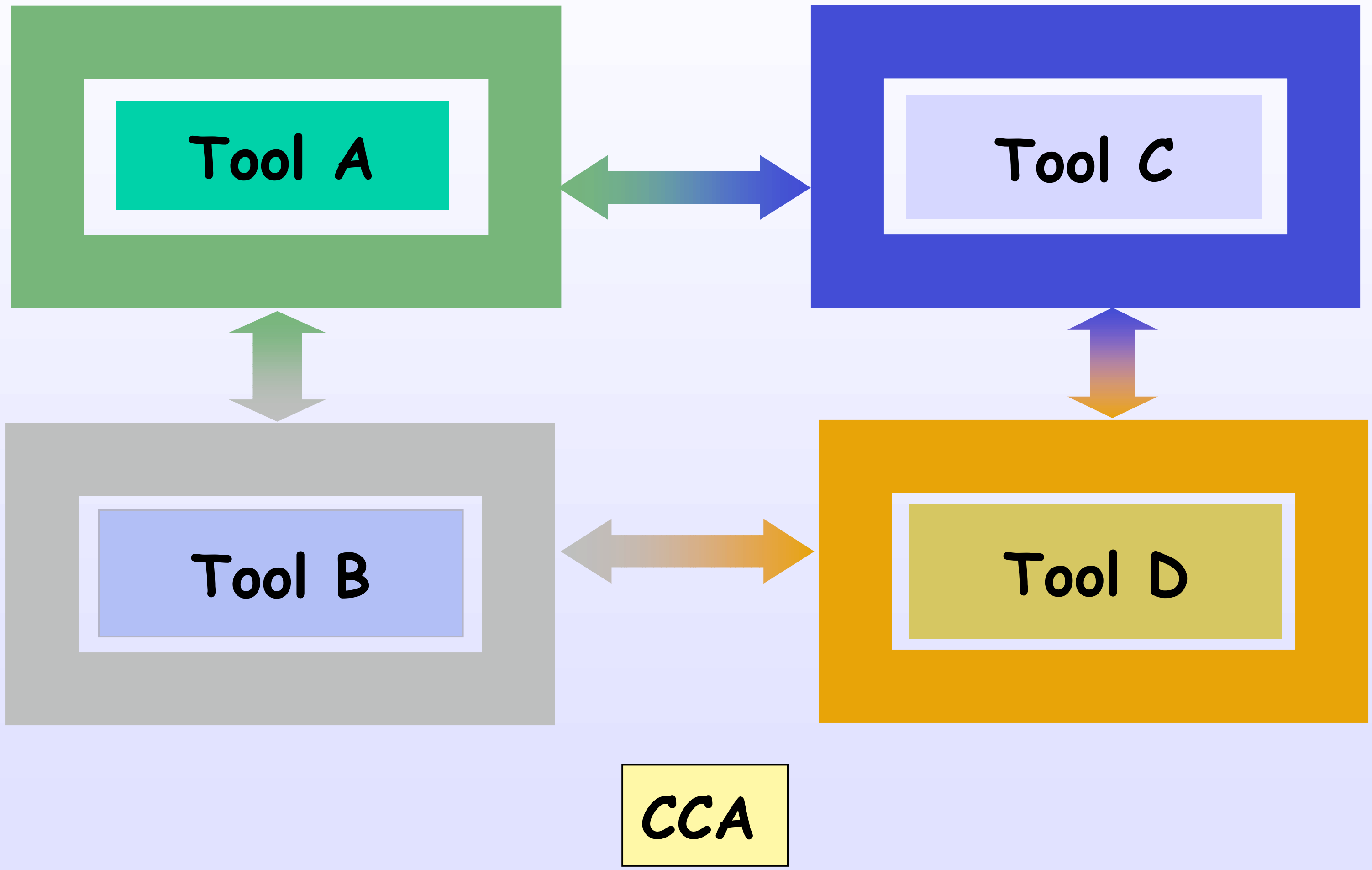
Globus

TAU

Tool Interoperability
Tool-to-Tool



Component Technology!



PSE's and Frameworks

PMatlab

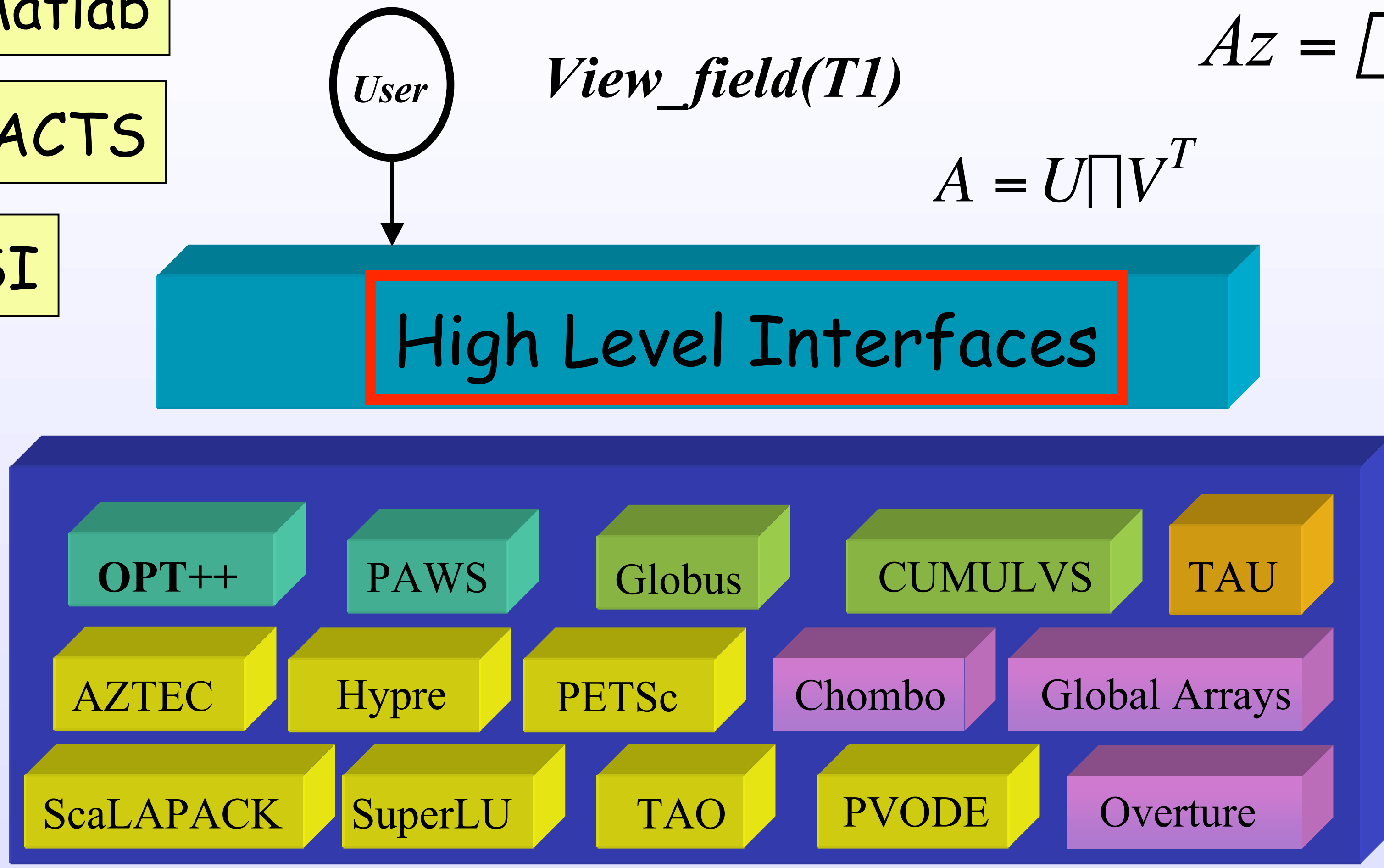
PyACTS

ESI

$$Ax = b$$

$$Az = \lambda z$$

$$A = U\Sigma V^T$$



PyACTS

Terminal

Window Edit Options Help

```
% run 4

---------------------------------------------------------------
 User: nkang                         Repo: mpccc
 Job Name: <none specified>          Group: mpccc
 Class Of Service: interactive       Job Class: interactive
 Job Accepted: Wed Jul 30 09:42:16 2003
---------------------------------------------------------------

llsubmit: Processed command file through Submit Filter: "/usr/common/nsg/etc/sub
filter".
>>>
import sys
>>>
sys.path.append('/u6/nkang/kn/pyacts_1/build/lib.aix-5.1-mpi-2.2')
>>>
import scalapack
>>>
scalapack.ex2("ex2_mat","ex2_rhs","sol",6,1,2,2,2,1)

Scalapack Example Program #2 (C-version) -- 07/24/2003
Solving AX=B
where A is a 6 by 6 matrix,
B is a 6 by 1 matrix,
with a block size of 2
Running on 4 processes, where the process grid is 2 by 2
INFO code returned by PDGESV = 0

According to the normalized residual the solution is correct.
||AX-B|| / (||X||*||A||*eps*N) =   1.25878215e-01

The solution is written to file sol

End of test.
>>>
```

This weeks agenda!

Agenda

<table>
<tr><th>Tuesday Aug 5</th><th>Wednesday Aug 6</th><th>Thursday Aug 7</th><th>Friday Aug 8</th></tr>
<tr><td rowspan="3">Introduction to Computational Environments</td><td rowspan="2">Invited Talk</td><td>Invited Talk</td><td rowspan="3">Support for Computational Environments</td></tr>
<tr><td>Numerical Optimization</td></tr>
<tr><td>Solution of Linear Systems (direct) and Eigenvalue problems</td><td rowspan="2">Numerical Grid/Mesh Manipulation</td></tr>
<tr><td rowspan="2">Support for PDEs</td><td rowspan="2">Support for PDEs</td><td rowspan="3">CCA</td></tr>
<tr><td rowspan="2">Support for Computational Environments</td></tr>
<tr><td rowspan="2">Numerical Optimization</td><td rowspan="2">Grid Mesh/Mesh Manipulation</td></tr>
<tr><td>Remote Stearing and Visualization</td><td>Performance And Tuning</td></tr>
</table>

acts-support@nersc.gov

http://acts.nersc.gov